FLIGHT PRICE PREDICTION

Guided by,
Shwetha Bedarkar

SUBMITTED BY,
HANSHILA AND ANUSHA

OUTLINE

- ABSTRACT
- PROBLEM STATEMENT
- PROJECT SPECIFICATION
- DATASET DESCRIPTION
- PIPELINE
- DATA MINING
- DATA CLEANING
- EXPLORATORY DATA ANALYSIS
- DATA VISUALISATION
- FEATURE ENGINEERING
- MODEL EVALUATION
- RESULT & OUTPUT
- CONCLUSION

ABSTRACT

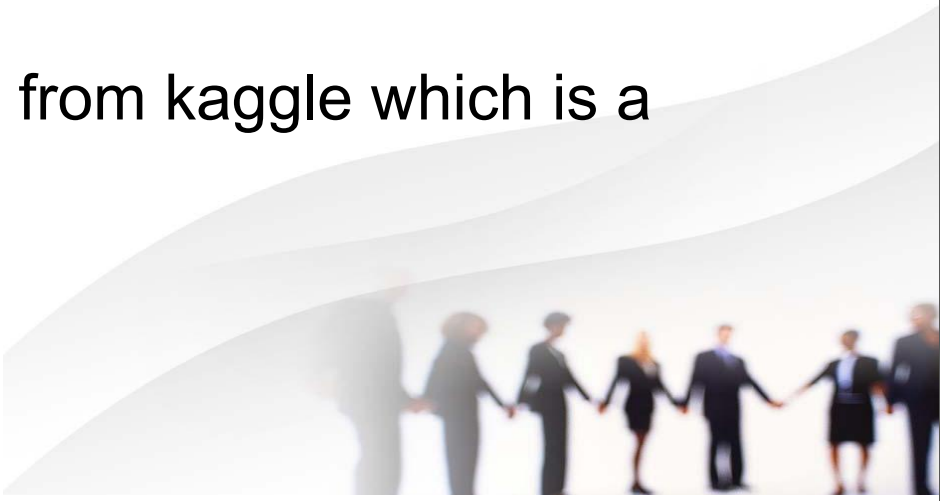
- We developed a machine learning model to predict the price of the flight.
- This model can provide the ticket prices of every flight tickets.
- It is a platform that is extremely benefical for the passengers and airlines.
- It is a model with high accuracy score which is created using the XGB Regressor Algorithm.
- A thorogh study of the data will aid in the discovery of valuable insights that wil be of enormous value to passengers and airlines.

PROBLEM STATEMENT

- Nowadays, the number of people using flights has increased significantly.
- It is difficult for airlines to maintain prices since price changes dynamically due to different conditions .
- Thus, the flight prices are something hard to predict.

PROJECT SPECIFICATION

- ✓ As a data Analyst , we are using various Machine Learning Models to solve this problem.
- ✓ This can help airlines by predicting what prices they can maintain .
- ✓ It can also help customers to predict future prices and their journey accordingly .
- ✓ The model is created by obtaining the information from kaggle which is a freely available platform .
- ✓ The data contains 10683 records of data .



DATASET DESCRIPTION

- ✓ The Dataset was obtained from kaggle.com.
- ✓ There are 10683 observation in our dataset with 11 columns .

1. **Airline** : the name of the airline.
2. **Date_of_journey**: the date of the journey.
3. **Source**: the souce from which the service begins.
4. **Destination**: the destination where the service ends.
5. **Route**: the route taken by the flight to reach the destination.
6. **Dep_time**: the time when you starts from the source.
7. **Arrival_time**:time of arrival of destination.
8. **Duration**:total duration of the flight
9. **Total_stops**:total stops between the source and destination.
10. **Additional_info**:additional information about the flight.
11. **price**: the price of the ticket

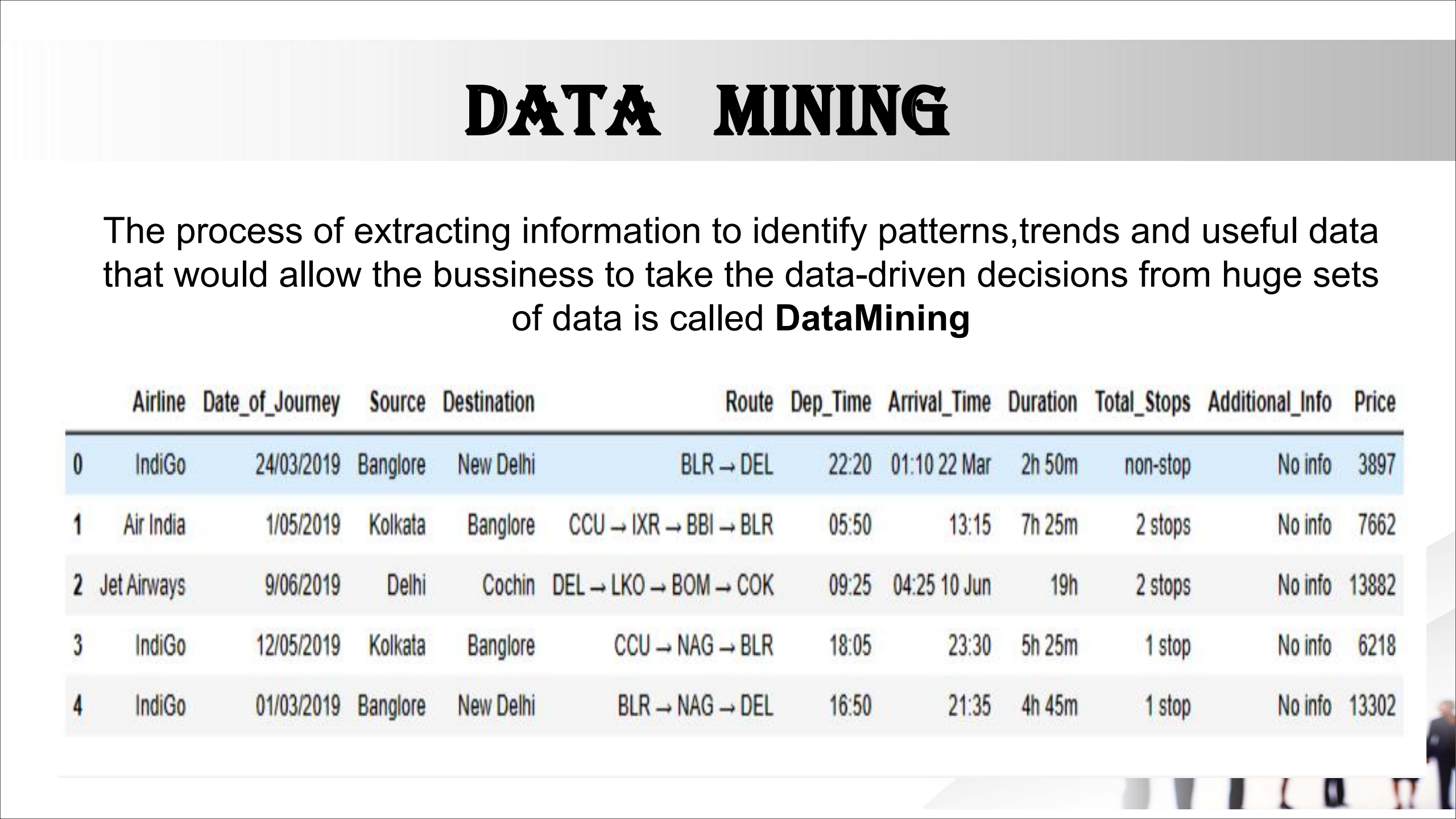
PIPELINE

- DATA MINING
- DATA CLEANING
- EXPLORATORY DATA ANALYSIS
- DATA VISUALISATION
- MODEL EVALUATION

DATA MINING

The process of extracting information to identify patterns,trends and useful data that would allow the bussiness to take the data-driven decisions from huge sets of data is called **DataMining**

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302



DATA MINING

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302

```
df.shape
```

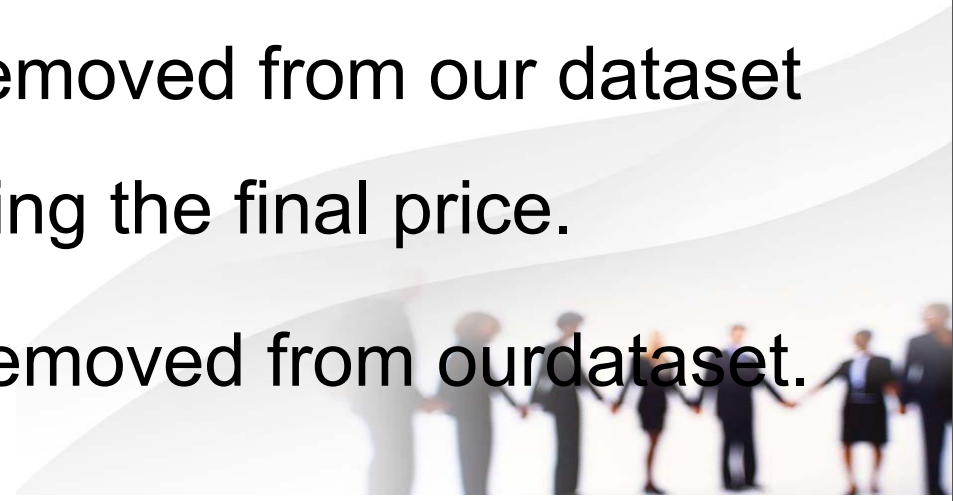
```
(10683, 11)
```

```
df.columns
```

```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info', 'Price'],
      dtype='object')
```

DATA CLEANING

- The main goal of the data cleaning is to identify and remove errors and duplicate data in order to create a trustworthy dataset.
- pandas , is a well-known library programme ,is used in the data cleaning process.
- Those columns and features are initially removed from our dataset because they are unimportant in determining the final price.
- Rows with null values in any column are removed from ourdataset.

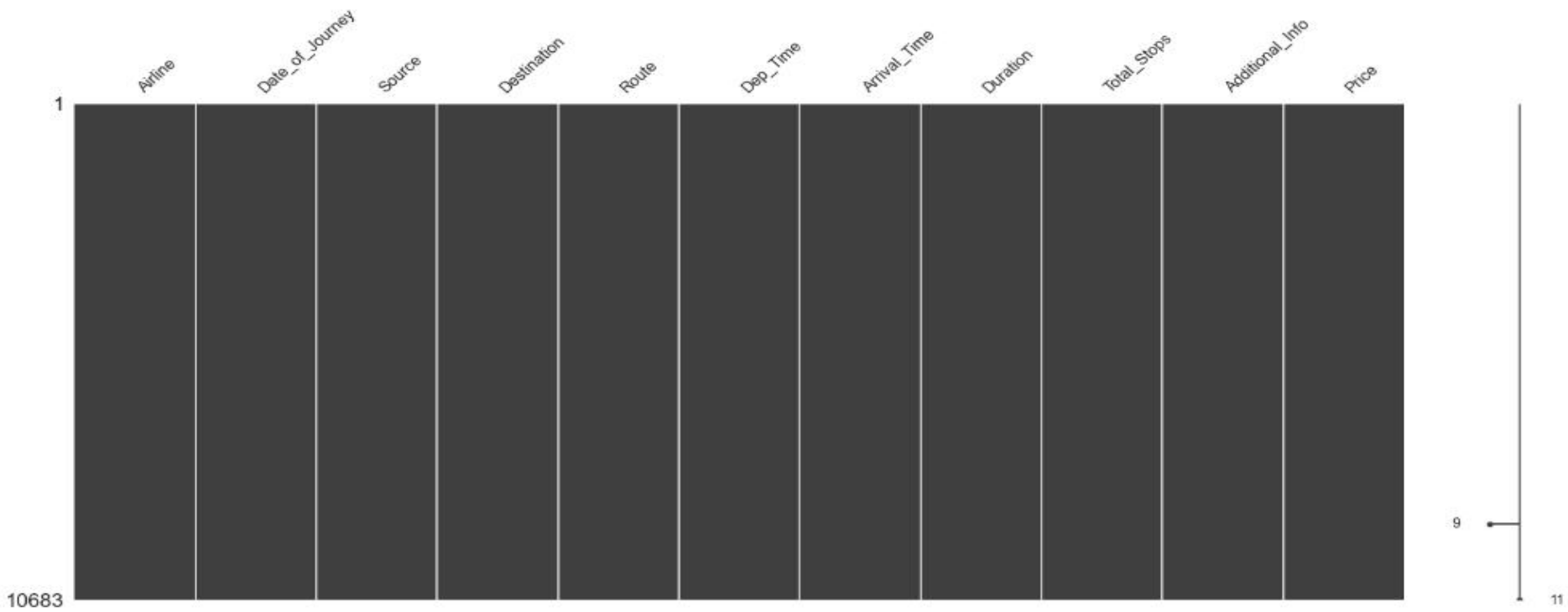


DATA CLEANING

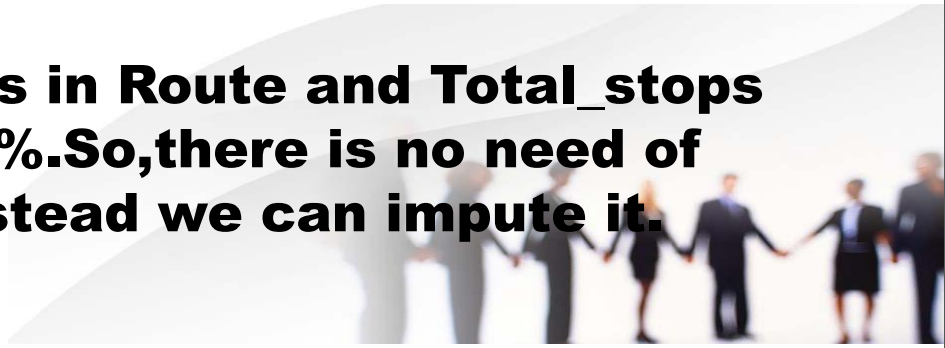
```
#checking the missing value
df.isnull().sum()/len(df)*1
```

Airline	0.000000
Date_of_Journey	0.000000
Source	0.000000
Destination	0.000000
Route	0.009361
Dep_Time	0.000000
Arrival_Time	0.000000
Duration	0.000000
Total_Stops	0.009361
Additional_Info	0.000000
Price	0.000000

dtype: float64

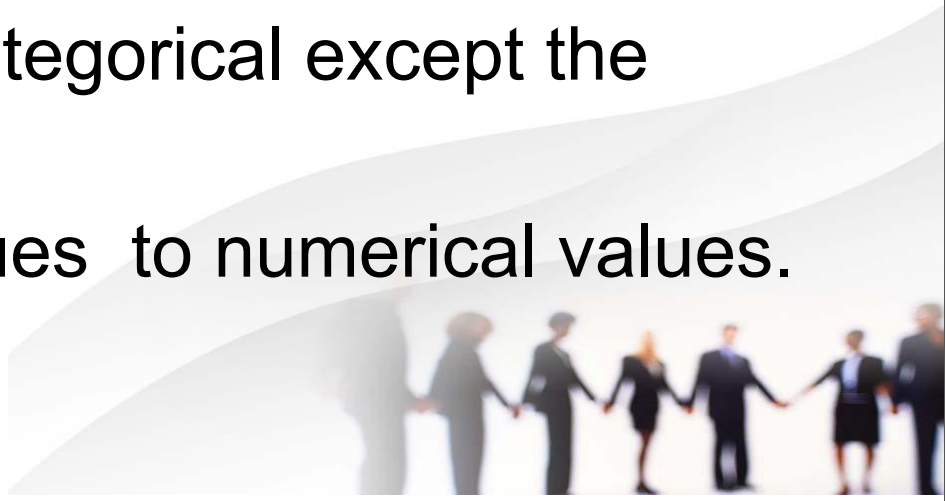


We can observe missing values in Route and Total_stops features which is less than 30%.So,there is no need of dropping the entire column instead we can impute it.



EXPLORATORY DATA ANALYSIS

- EDA is applied to investigate the data and summarise the key insights.
- It will give you the basic understanding of your data , its distribution, null values and much more.
- you can either explore data using graphs or through some python functions .
- In this all the independent variables are categorical except the price which is target variable .
- So, we are converting the categorical values to numerical values.



EXPLORATORY DATA ANALYSIS

```
duration = list(df["Duration"])
for i in range(len(duration)):
    if len(duration[i].split()) != 2:    # Check if duration contains only hour or mins
        if "h" in duration[i]:
            duration[i] = duration[i].strip() + " 0m"   # Adds 0 minute
#             print('Hour ',duration)
        else:
            duration[i] = "0h " + duration[i]          # Adds 0 hour
#             print('Minutes \n',duration)
```

```
duration_hours = []
duration_mins = []
for i in range(len(duration)):
    duration_hours.append(int(duration[i].split(sep = "h")[0]))    # Extract hours from duration
    duration_mins.append(int(duration[i].split(sep = "m")[0].split()[-1]))   # Extracts only minutes from duration
```

```
df["Duration_hours"] = duration_hours
df["Duration_mins"] = duration_mins
```

```
df["Journey_day"]=pd.to_datetime(df["Date_of_Journey"], format="%d/%m/%Y").dt.day
```

```
df["Journey_month"] = pd.to_datetime(df["Date_of_Journey"], format = "%d/%m/%Y").dt.month
```

```
df["Dep_hour"]=pd.to_datetime(df["Dep_Time"]).dt.hour
```

```
df["Dep_min"]=pd.to_datetime(df["Dep_Time"]).dt.minute
```

```
df["Arrival_hour"] = pd.to_datetime(df["Arrival_Time"]).dt.hour
df["Arrival_min"] = pd.to_datetime(df["Arrival_Time"]).dt.minute
```

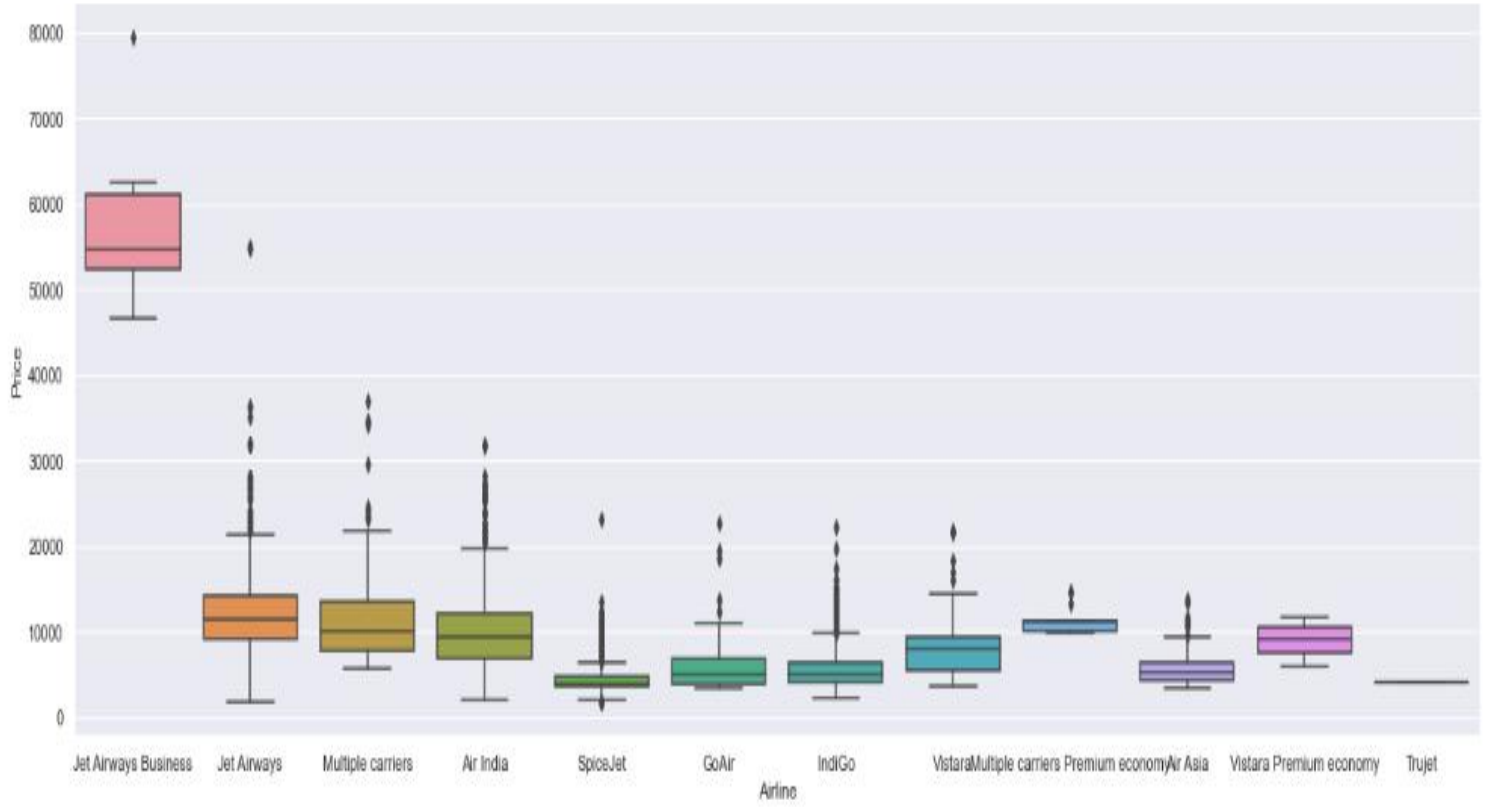
DATA VISUALISATION

AIRLINE VS PRICE

Observations:

- Jet Airways has high price as compared to other flights.
- Remaining Airways have almost same prices.
- There are high outliers in jet Airways.

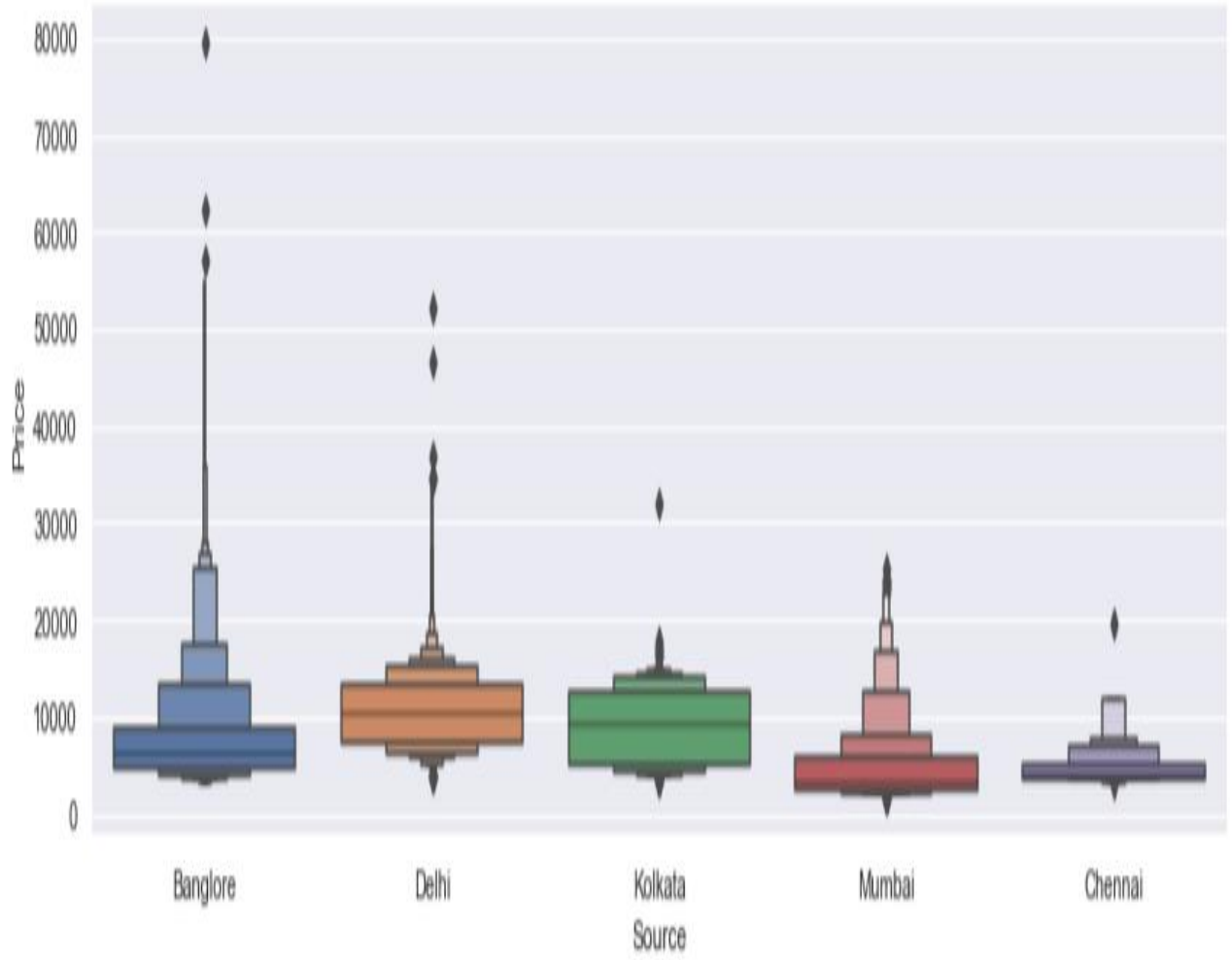
```
#plot of Airline vs price
sns.catplot(y = "Price", x = "Airline", data = df.sort_values("Price", ascending = False), kind="box", height = 6, aspect = 3)
plt.show()
```



SOURCE VS PRICE

- Chennai value in source column has less flight price as compared to rest of the values.
- Banglore shows a higher flight price.

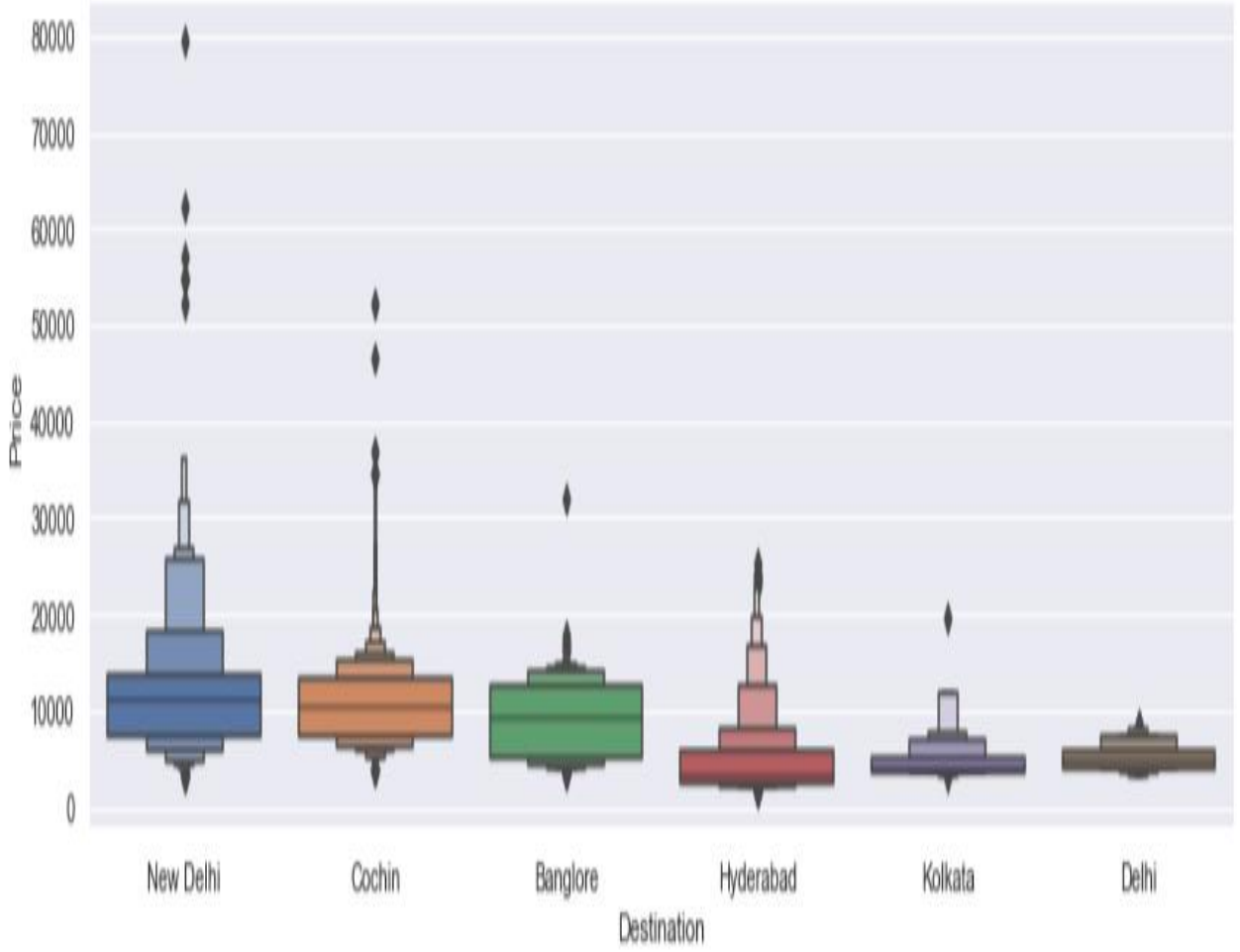
```
#plot of source vs price
sns.catplot(y = "Price", x = "Source", data = df.sort_values("Price", ascending = False), kind="boxen", height = 4, aspect = 3)
plt.show()
```



DESTINATION VS PRICE

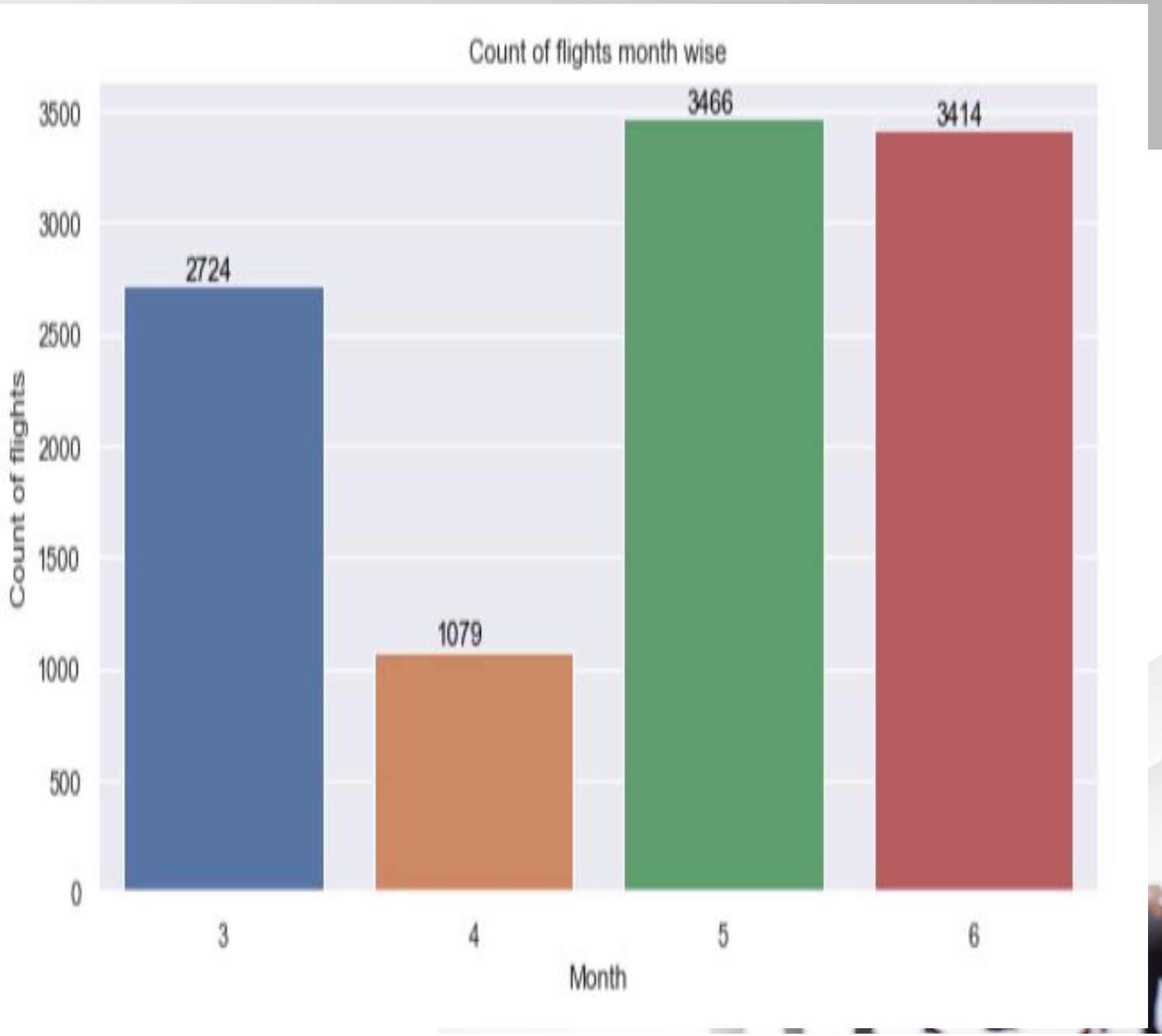
- Here we are plotting the box plot with the help of catplot between price of the flight and destination to which passenger is travelling to and figured out that New delhi has the most outliers and Kolkata has the least.

```
#plot of destination vs price
sns.catplot(y = "Price", x = "Destination", data = df.sort_values("Price", ascending = False), kind="boxen", height = 4, aspect =
plt.show()
```



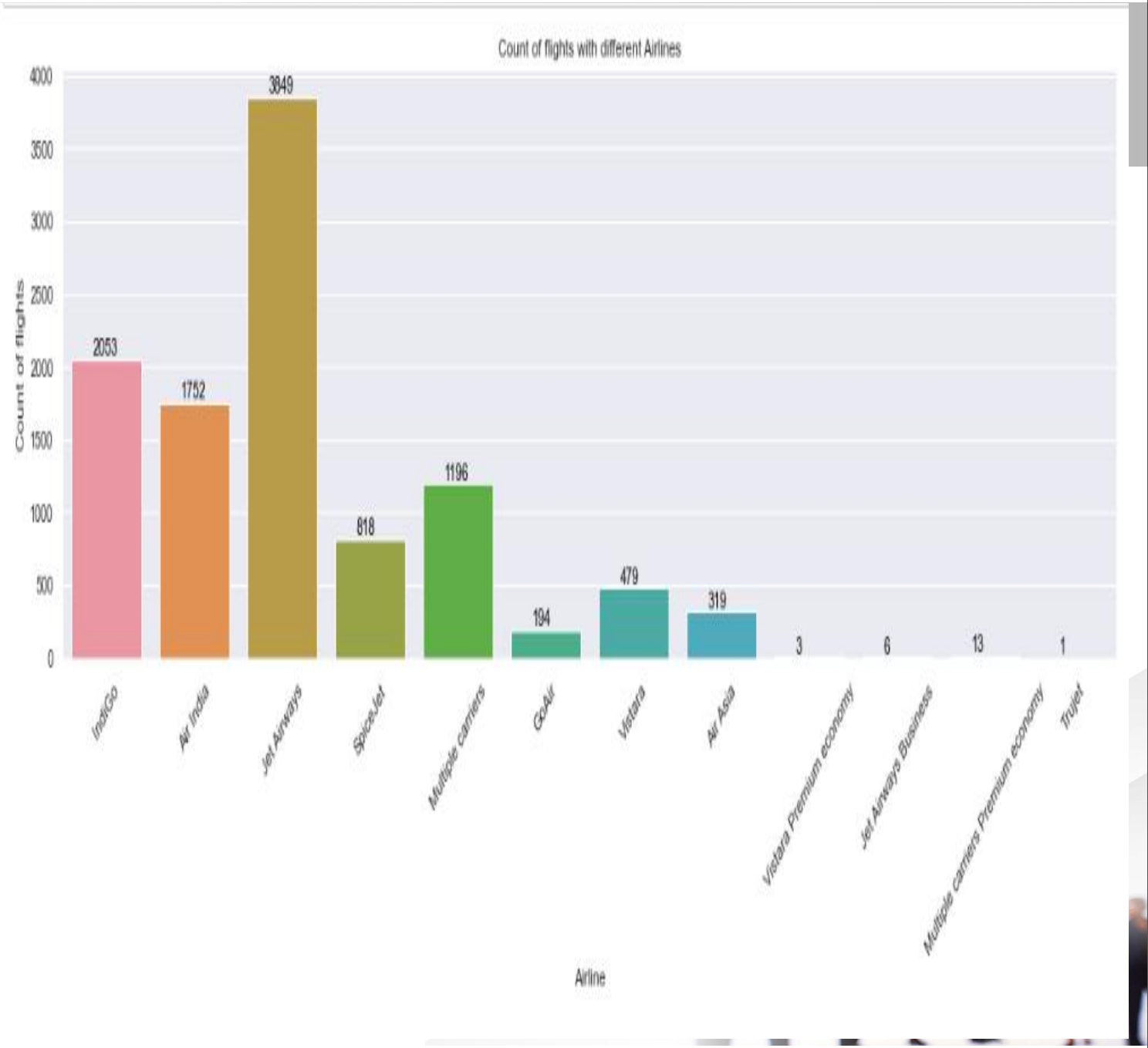
MONTHS VS NO.OF FLIGHTS

- Here in the above graph we have plotted the count plot for journey in month vs number of flights and got to see that May has the most number of flights.



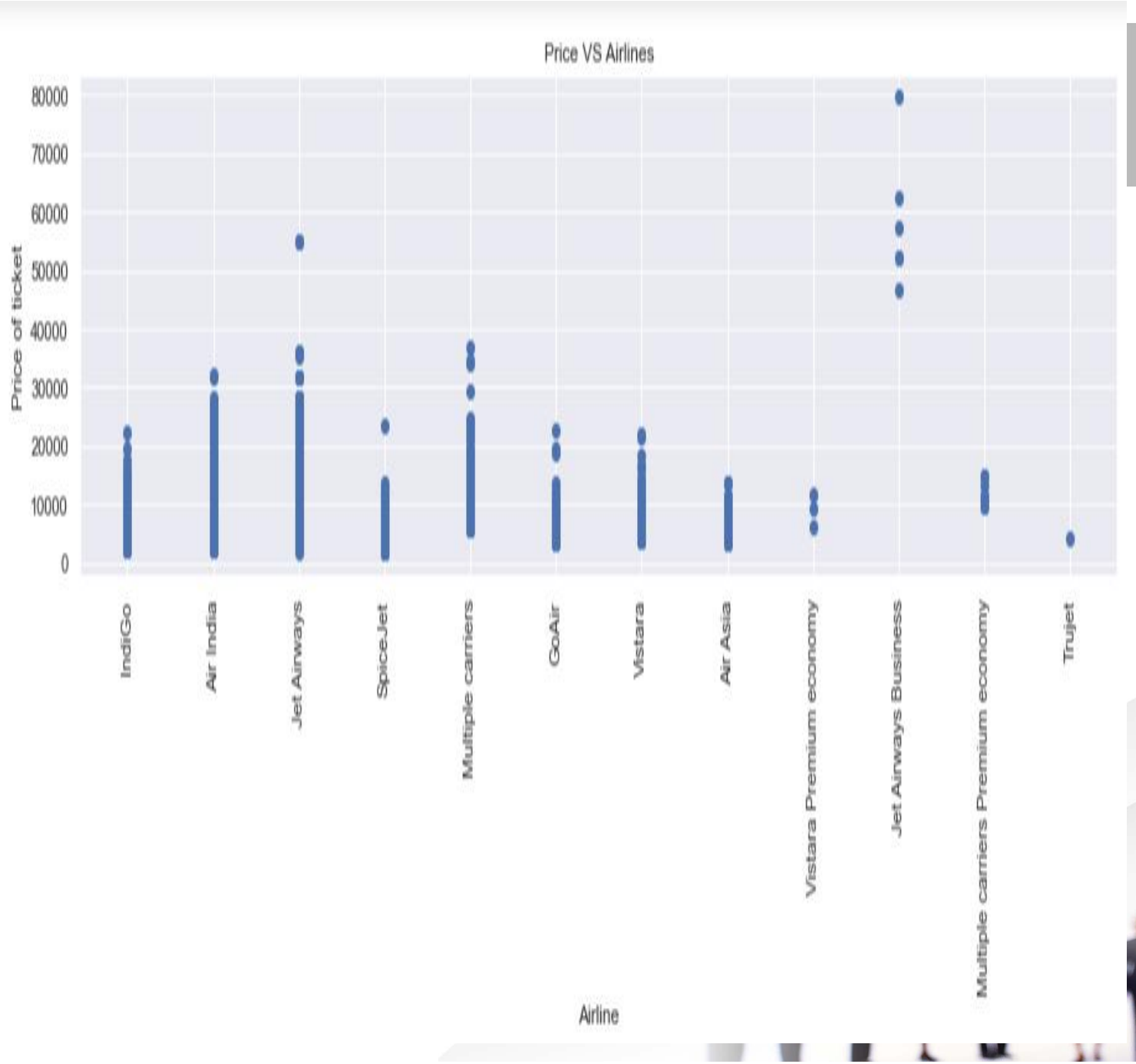
TYPES OF AIRLINES VS NO.OF FLIGHTS

- From the above graph we can see that between the type of airlines and count of flights.
- we can see that Jet airways has the most flight boarded.



TICKET PRICES VS AIRLINES

- Here we can observe that Jet Airways Business has highest ticket price and Trujet is having less price.



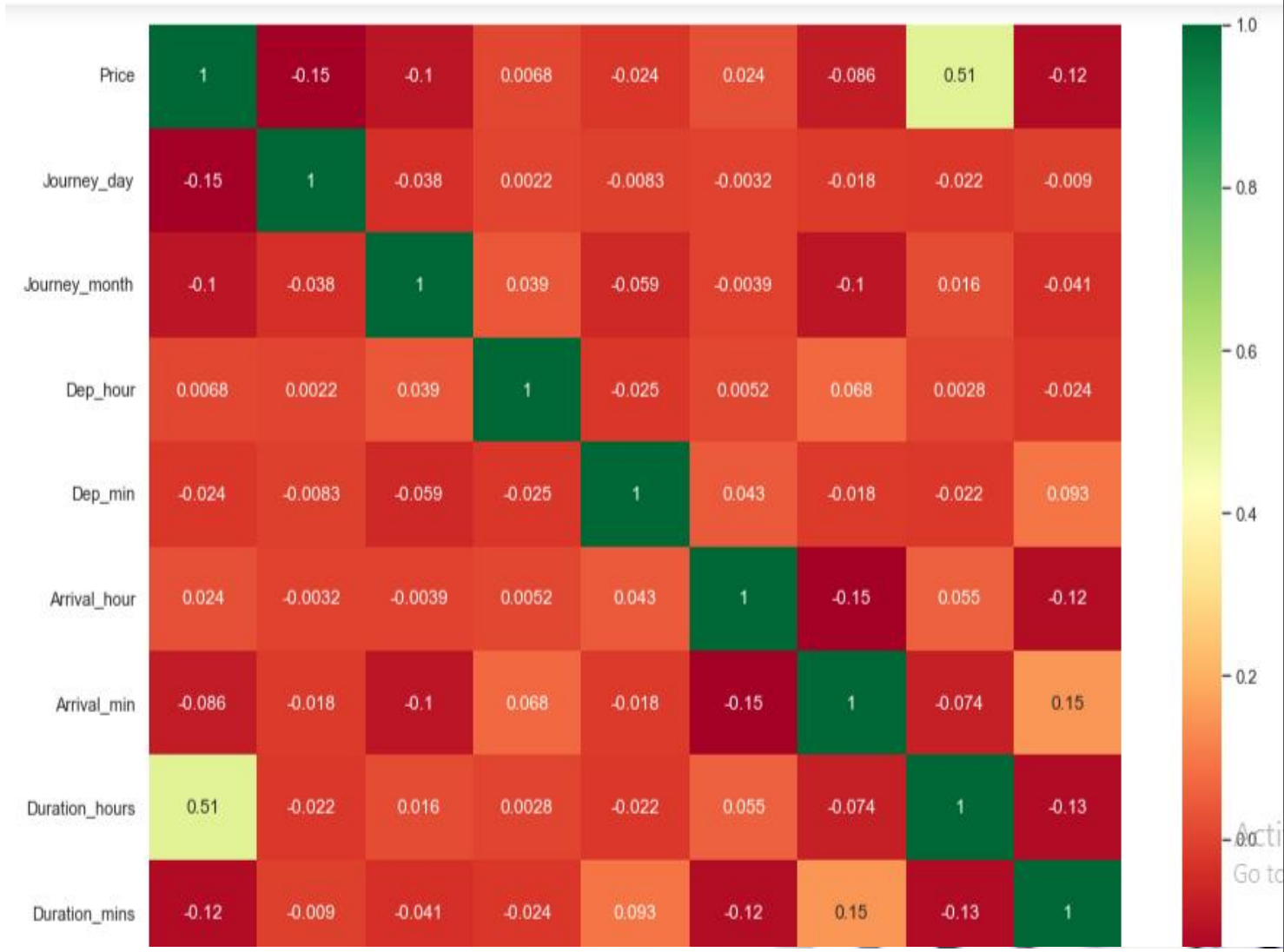
FEATURE ENGINEERING

- Feature engineering is the process of extracting features from raw data using domain knowledge and data mining techniques. these characteristics can help machine learning algorithms perform better.
- Feature engineering can be thought of as applied machine learning.

FEATURE SELECTION

1. Heat Maps

- Heat maps are very useful to find relation between two variables in a data set.
- Here heat maps are used to find how the variable are dependent to each other.



2.**Lasso Technique**

- Lasso regression algorithm is defined as a regularization algorithm that assists in the elimination of irrelevant parameters, thus helping in the concentration of selection and regularizes the models.
- The LASSO method regularizes model parameters by shrinking the regression coefficients, reducing some of them to zero. The feature selection phase occurs after the shrinkage, where every non-zero value is selected to be used in the model.

	Importance	Columns
0	2694.063705	Total_Stops
1	-76.668639	Journey_day
2	-410.982083	Journey_month
3	-13.199133	Arrival_hour
4	1369.951746	Airline_Air India
5	-0.000000	Airline_GoAir
6	10.709160	Airline_IndiGo
7	4107.885940	Airline_Jet Airways
8	50070.257209	Airline_Jet Airways Business
9	3425.657489	Airline_Multiple carriers
10	2898.522386	Airline_Multiple carriers Premium economy
11	-490.219854	Airline_SpiceJet
12	1864.398368	Airline_Vistara
13	0.000000	Airline_Vistara Premium economy
14	0.000000	Source_Chennai
15	94.210507	Source_Delhi
16	-0.000000	Source_Kolkata
17	-1647.643434	Source_Mumbai
18	0.000000	Destination_Cochin
19	-873.149298	Destination_Delhi
20	-111.058463	Destination_Hyderabad
21	13.180554	Destination_Kolkata
22	1766.985193	Destination_New Delhi
23	-0.708044	Duration_mins
24	6.280317	Duration_hours

FEATURE ENCODING

- Machine learning models cn only work with numerical values.
- For this reason,it is neccessary to transform the categorical values of the relevant features into numerical ones.
- This process is called feature encoding

➢ Feature encoding techniques are:

1. One hot encoder
2. Label encoder

1.ONE HOT ENCODER

	Airline_Air India	Airline_GoAir	Airline_IndiGo	Airline_Jet Airways	Airline_Jet Airways Business	Airline_Multiple carriers	Airline_Multiple carriers Premium economy	Airline_SpiceJet	Airline_Vistara	Airline_Vistara Premium economy
0	0	0	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0

```
#Next we have Source as nominal data to convert
Source=df[['Source']]
Source=pd.get_dummies(Source,drop_first=True)
Source.head()
```

	Source_Chennai	Source_Delhi	Source_Kolkata	Source_Mumbai
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	0

2.LABEL ENCODER

```
df["Total_Stops"].replace({"non-stop":0,"1 stop":1,"2 stops":2,"3 stops":3,"4 stops":4},inplace=True)
df.head()
```

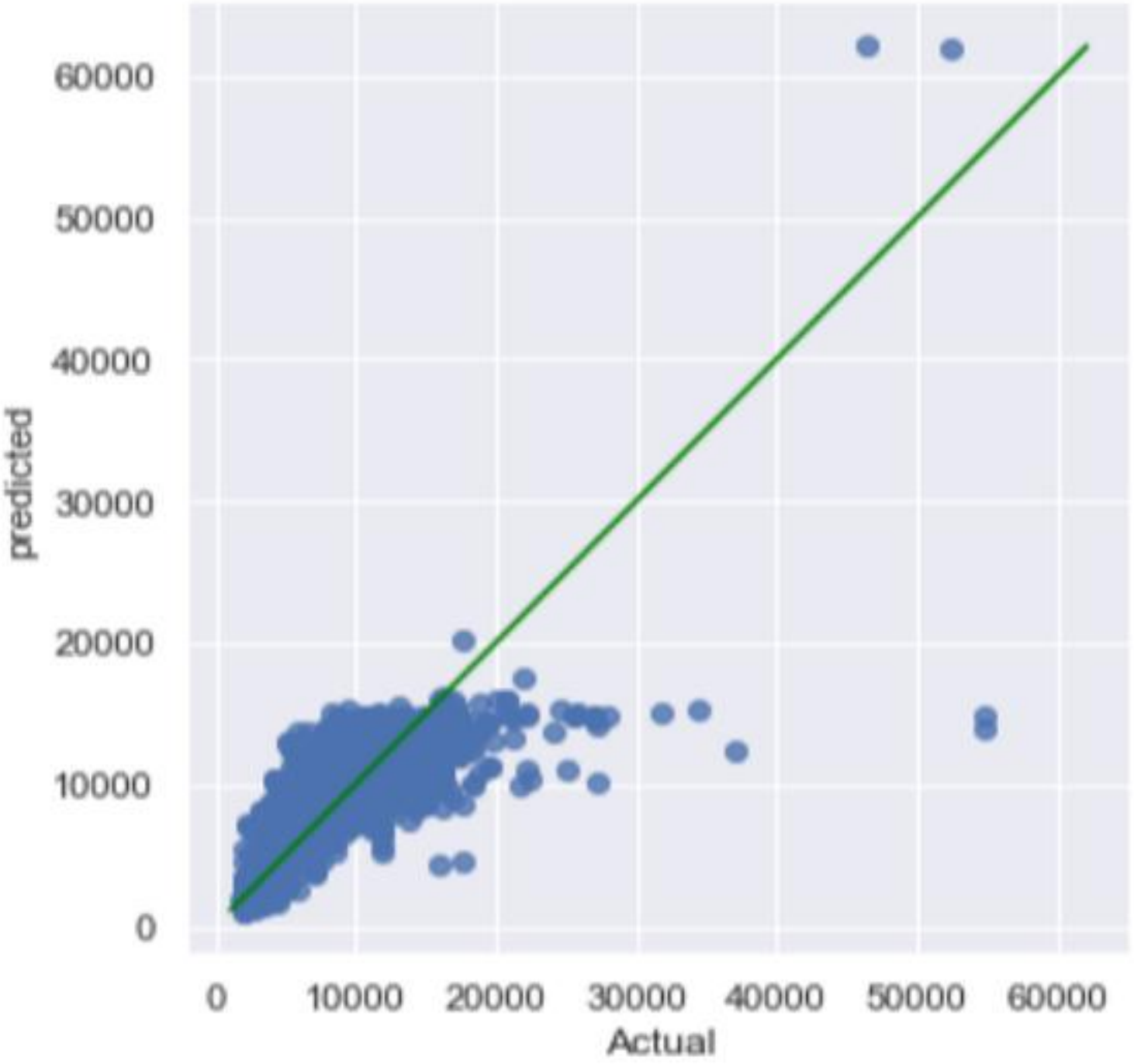
	Airline	Source	Destination	Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Duration_hours	Duration_mins
0	IndiGo	Banglore	New Delhi	0	3897	24	3	22	20	1	2	50
1	Air India	Kolkata	Banglore	2	7662	1	5	5	50	13	7	25
2	Jet Airways	Delhi	Cochin	2	13882	9	6	9	25	4	19	0
3	IndiGo	Kolkata	Banglore	1	6218	12	5	18	5	23	5	25
4	IndiGo	Banglore	New Delhi	1	13302	1	3	16	50	21	4	45

MODEL EVALUATION

- Modelling is the process of training a machine learning algorithm to predict tables based on features.
- we have 80% training data 20% testing data .
- we used Linear Regression,Gradient Boosting Regession ,XGB Regression,Random Forest Regression,K- Neighbours Regression algorithm to train and test the model.
- our model has a 93% accuracy rate, which is quite good.

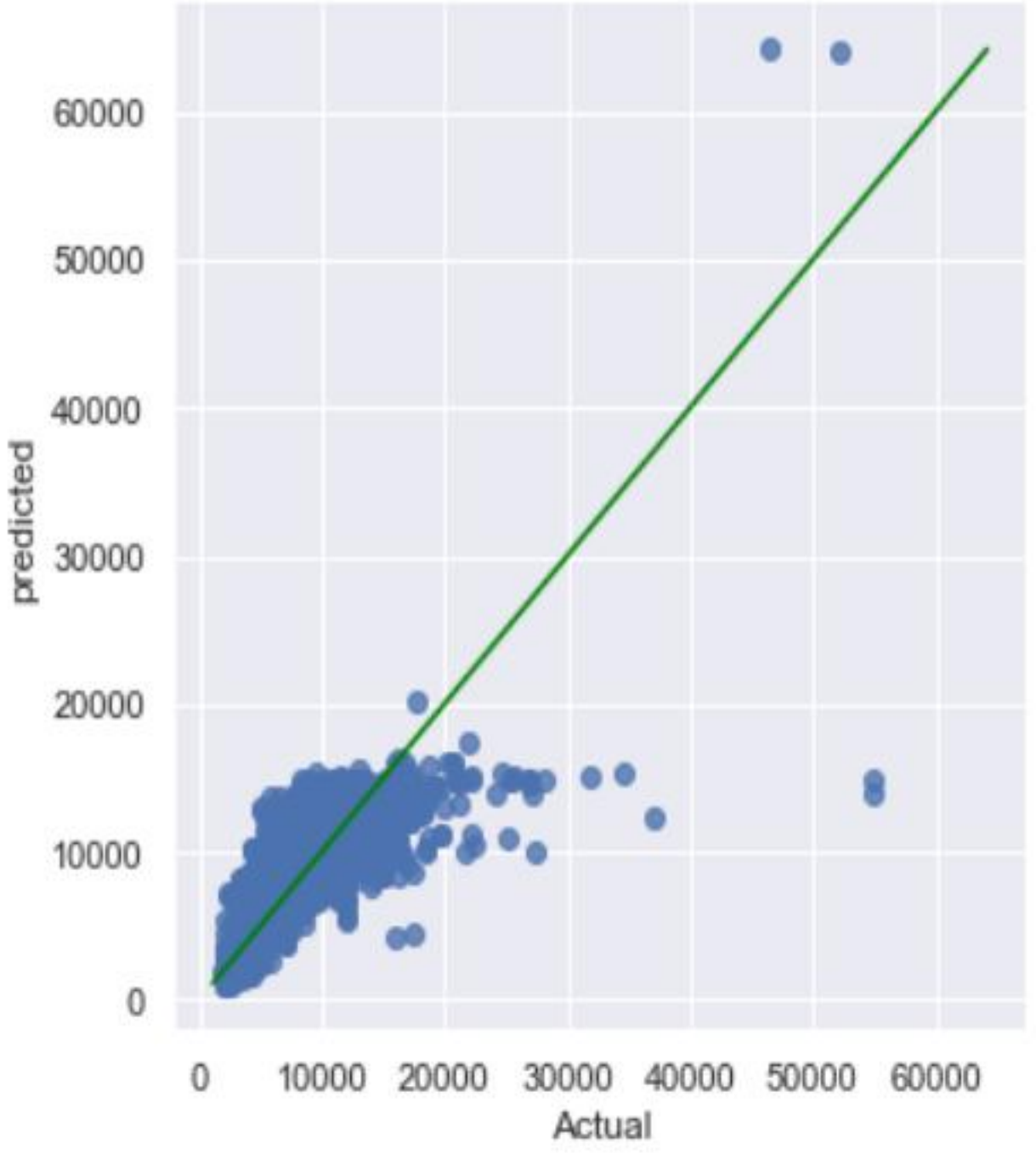
LASSO REGRESSION

R2 SCORE:0.68
Mean Squared Error:9.531
Adjusted R2 Score:0.63



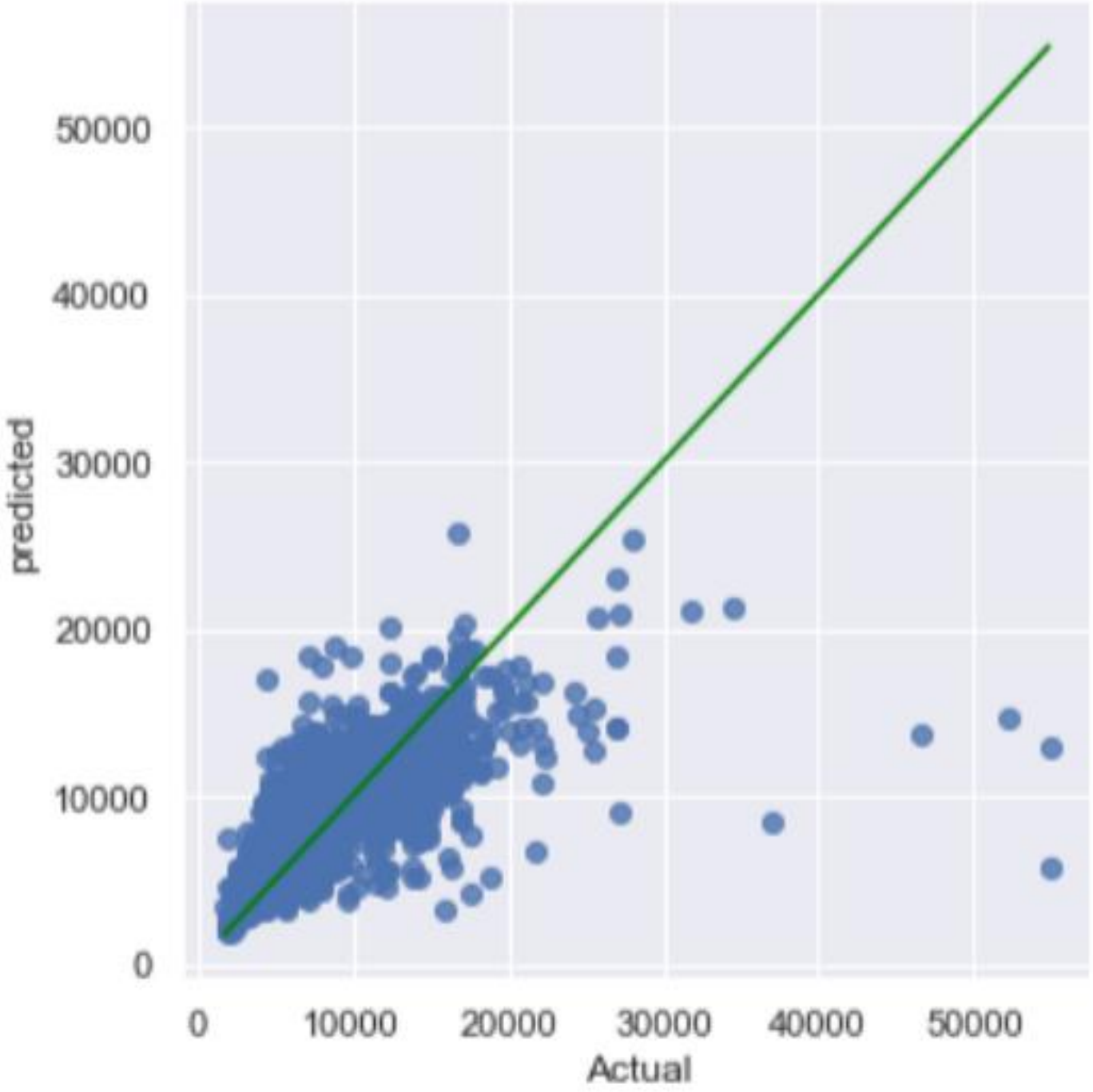
LINEAR REGRESSION

R2 SCORE : 0.63
Mean Squared Error : 9.57
Adjusted R2 Score : 0.63



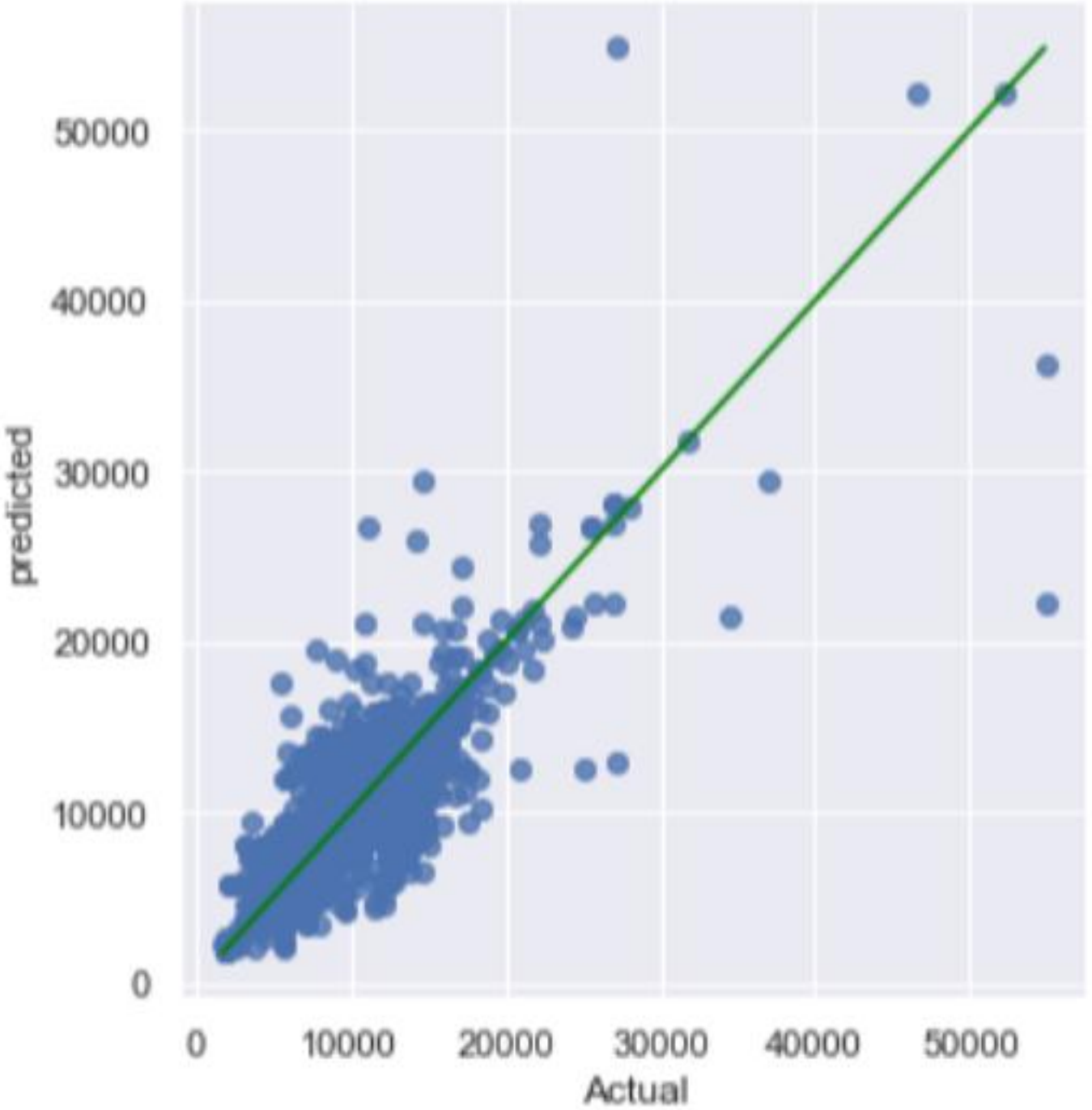
KNN REGRESSION

R2 SCORE : 0.73
Mean Squared Error : 1.061
Adjusted R2 Score : 0.73



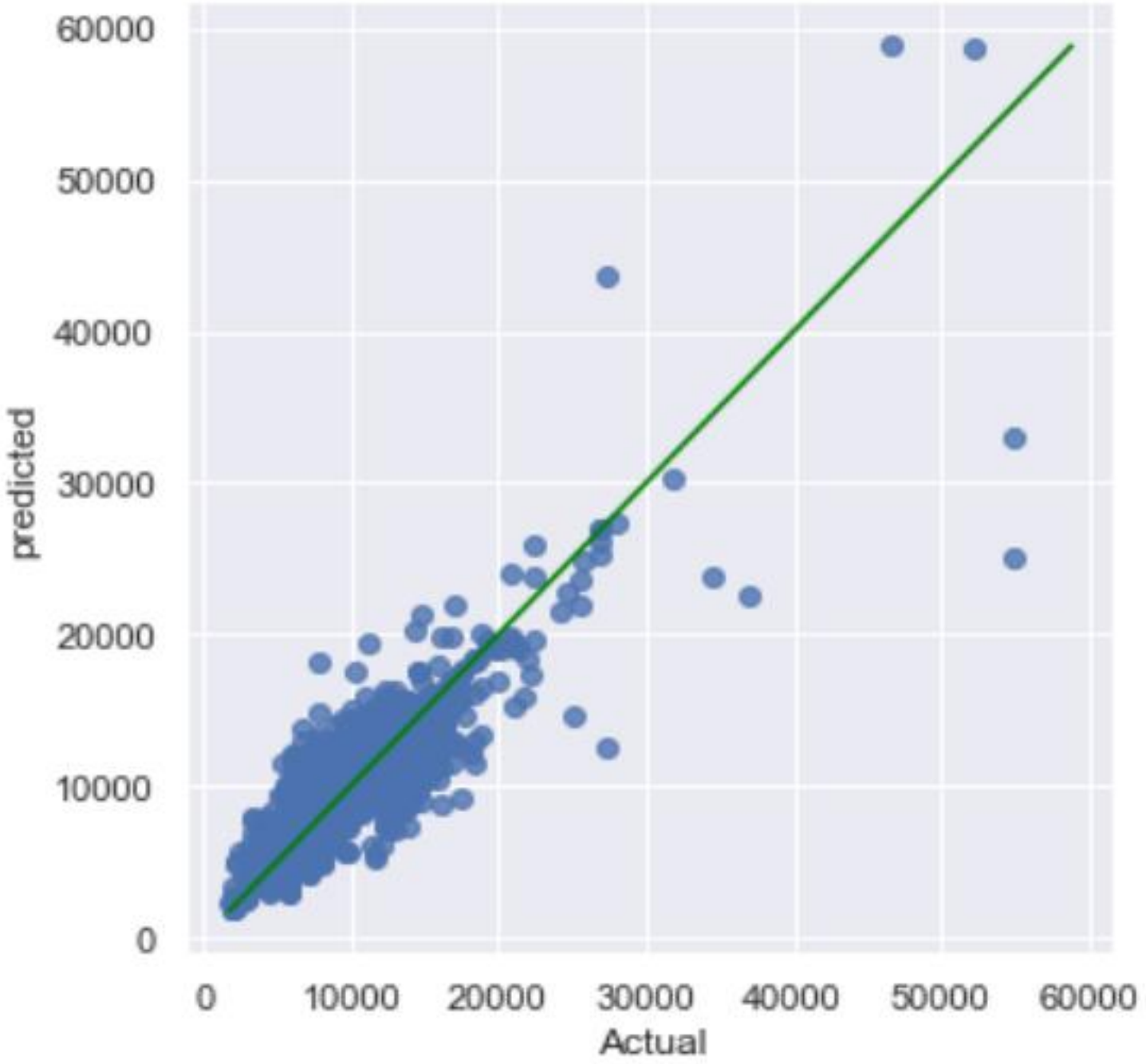
DECISION TREE REGRESSOR

R2 SCORE : 0.73
Mean Squared Error : 1.061
Adjusted R2 Score : 0.73



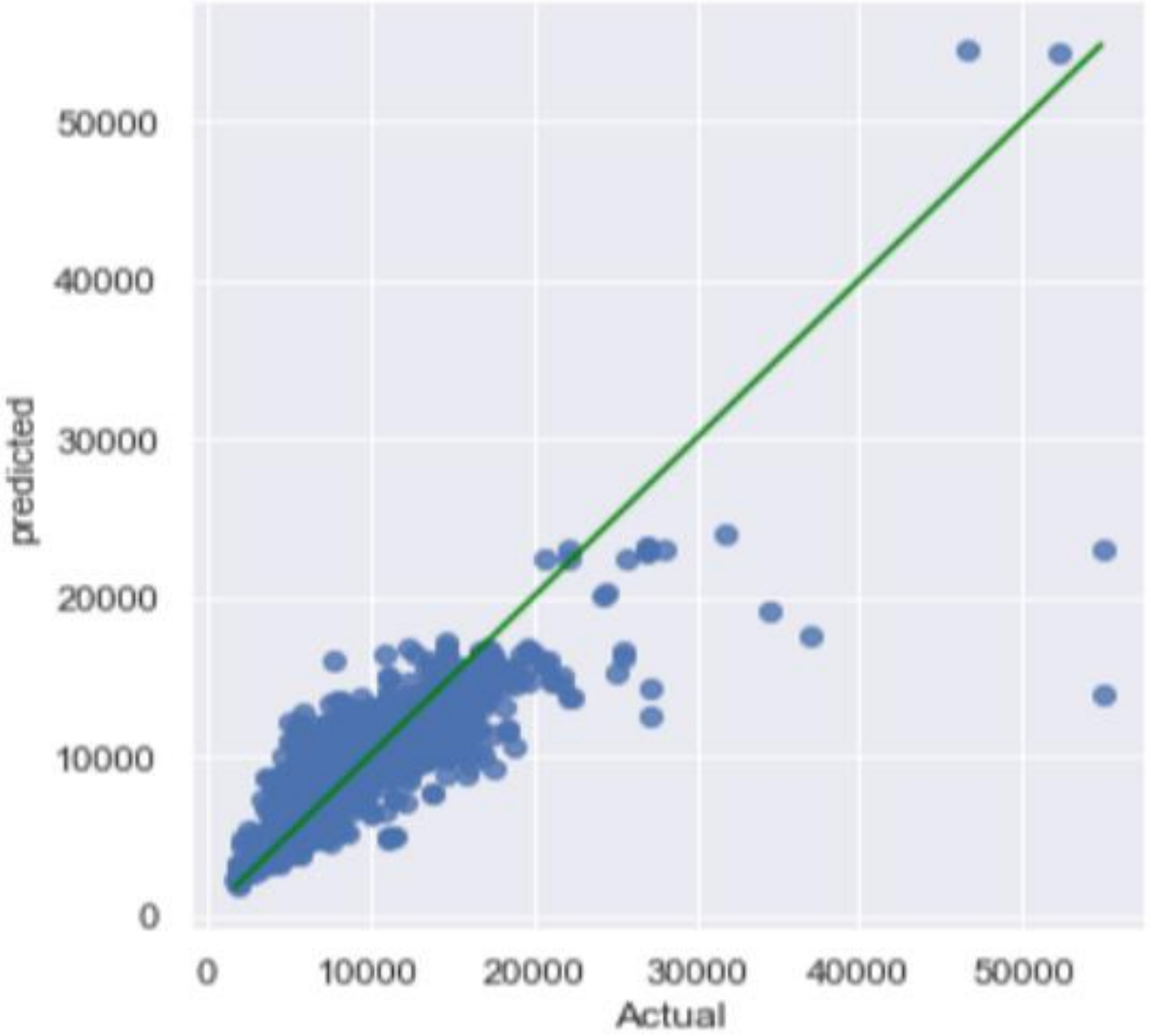
RANDOM FOREST REGRESSOR

R2 SCORE : 0.95
Mean Squared Error :4.35
Adjusted R2 Score : 0.95



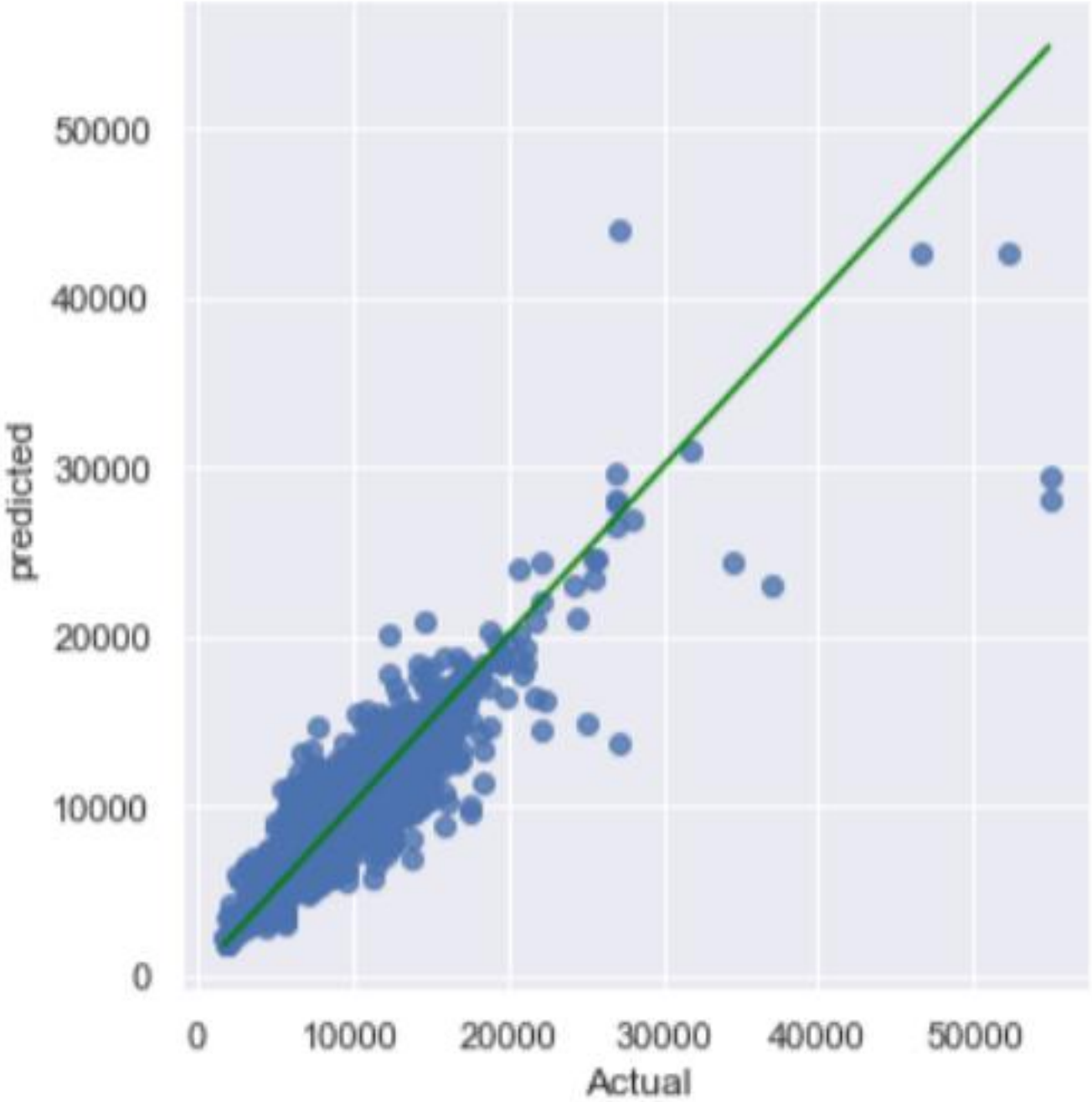
GRADIENT BOOSTER REGRESSOR

R2 SCORE : 0.78
Mean Squared Error :6.01
Adjusted R2 Score : 0.78



XGB REGRESSOR

R2 SCORE : 0.93
Mean Squared Error :3.62
Adjusted R2 Score : 0.93



IMPLEMENTATION

	ModelName	R-Square	Adj_R-Square	MSE	MAE	RMSE
0	LassoRegression	0.633306	0.632144	9.531019e+06	2061.030731	3087.234884
1	LinearRegression	0.633496	0.632506	9.578996e+06	2060.864733	3094.995377
2	KNN	0.735502	0.734789	1.061745e+07	1904.441086	3258.442354
3	DecisionTree	0.970751	0.970672	6.615420e+06	1429.372945	2572.045820
4	RandomForest	0.954644	0.954522	4.353456e+06	1247.595073	2086.493707
5	GB	0.789962	0.789395	6.011419e+06	1605.532974	2451.819605
6	XGB	0.932246	0.932063	3.627591e+06	1210.879136	1904.623517

MODEL PREDICTION

```
las_pred=lasso.predict(x_test)
las_pred
```

```
array([11238.67281305,  7994.16187774,  6290.6669745 , ...,
        7187.20700084, 11582.26328552,  2896.789707 86])
```

```
#Prediction on x_test and store it on y_pred
rf_pred=rf.predict(x_test)
rf_pred
```

```
array([12637.34116667,  4944.        ,  6340.18608333, ...,
        6557.43166667, 12395.0175    ,  3851.81066667])
```

```
kn_pred=knn.predict(x_test)
kn_pred
```

```
array([12586.6,  5419.6, 10542.2, ...,  6842.4, 12002.4,  3847.4])
```

```
dc_pred=dc.predict(x_test)
dc_pred
```

```
array([14781.,  4775.,  6144., ...,  7229., 14388.,  3857.])
```

```
gb_pred=gb.predict(x_test)
gb_pred
```

```
array([12218.92591182,  4288.53424127,  5157.4712198 , ...,
        6985.63857144, 11738.71123054,  4127.36554442])
```

```
lin_pred=lin.predict(x_test)
lin_pred
```

```
array([11241.19541424,  7997.03196038,  6291.42122464, ...,
        7197.89815028, 11586.02447888,  2885.17746753])
```

```
xg_pred =  xgb_reg.predict(x_test)
xg_pred
```

```
array([12570.336 ,  4804.953 ,  6000.3   , ...,  6467.159 , 11989.906 ,
        3864.6482], dtype=float32)
```

CONCLUSION

- In our project , the overall survey for the dynamic price changes in the flight tickets is presented . this gives the information about the highs and lows in the airfaires according to the days, weekends and time of the day.
- Also we use different reggresion models to predict the prices which are evaluated and compared in order to get better results.
- From the studies, we can suggest that more information like airline ticket payments details,number and placement of seat details ,covered auxiliary items and so on can create a more robust and complete flight price forecast model.

THANK YOU